Achtung Baustelle. Anleitung noch im Aufbau

Installation auf einem Rechner mit Ubuntu (z.B.: Linux Mint)

Die Installation auf einem Rechner mit Linux ist einfach und je nach Internetleitung innerhalb von 15 - 60 Minuten erledigt.

Vorraussetzung

PC oder Laptop mit einem aktuellen Linuxsystem.
Die Anleitung hier verwendet einen alten Laptop mit Linux Mint 22.1 Cinnamon

Schritt 1: Die ARDUINO IDE auf dem Linuxrechner installieren

Im Prinzip erfolgt die Installation der ARDUINO IDE auf dem Linuxrechner ähnlich der Installation auf einem Windows PC. Da es einige kleine aber wichtige Unterschiede gibt, findest Du im folgenden die angepasste Anleitung:

Installation ARDUINO IDE

Los geht’s

Als Einstieg und zum Ausprobieren reichen

- ein Arduino (Original oder preiswerter Clone)
- ein paar WS2812 RGB-LEDs und
- drei Kabel
- ein Rechner / Laptop mit Linux Mint 22.1 Cinnamon 64-Bit (32-Bit hat keine Unterstützung für Python 3.9) - Für einen 32-Bit Rechner/Laptop muß die LDME-Version von Mint installiert werden https://linuxmint.com/download_lmde.php. Diese Version enthält python 3.11.

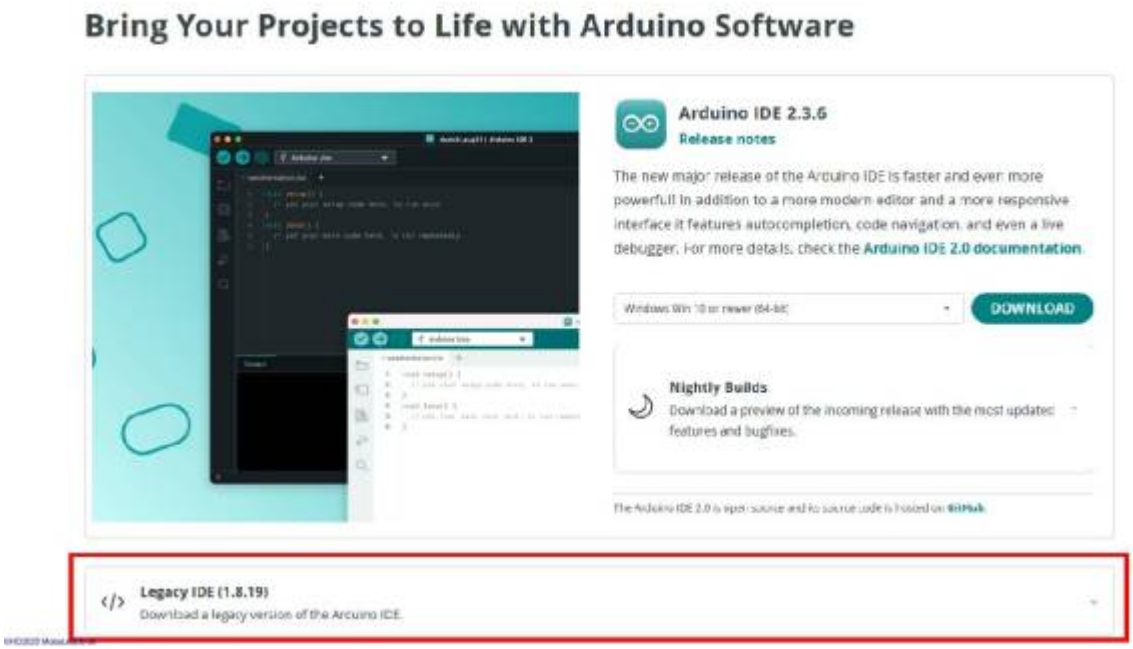
völlig aus. Der Arduino kann ein Uno, Nano oder alles, was die MLL unterstützt, sein.

Diese Anleitung gilt für eine Linux Umgebung mit einem installierten Linux Mint System.

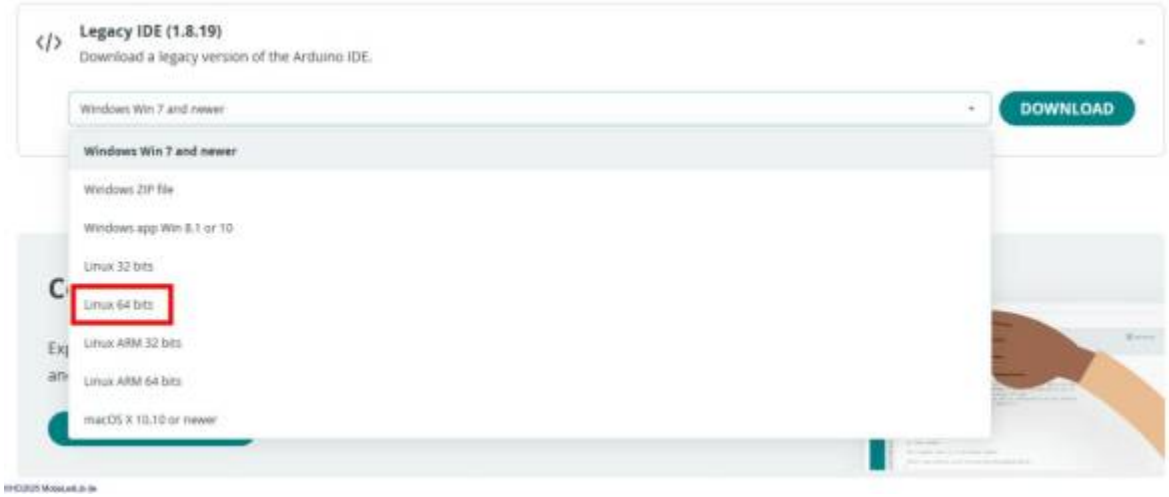
Zunächst muss man die aktuelle Arduino Entwicklungsumgebung IDE (Integrated Development Environment) herunterladen und installieren. Der Arduino WEB-Editor oder die Arduino App können für die MobaLedLib nicht genutzt werden.

Die ARDUINO Download Seite erreicht man über den Link: https://www.arduino.cc/en/Main/Software

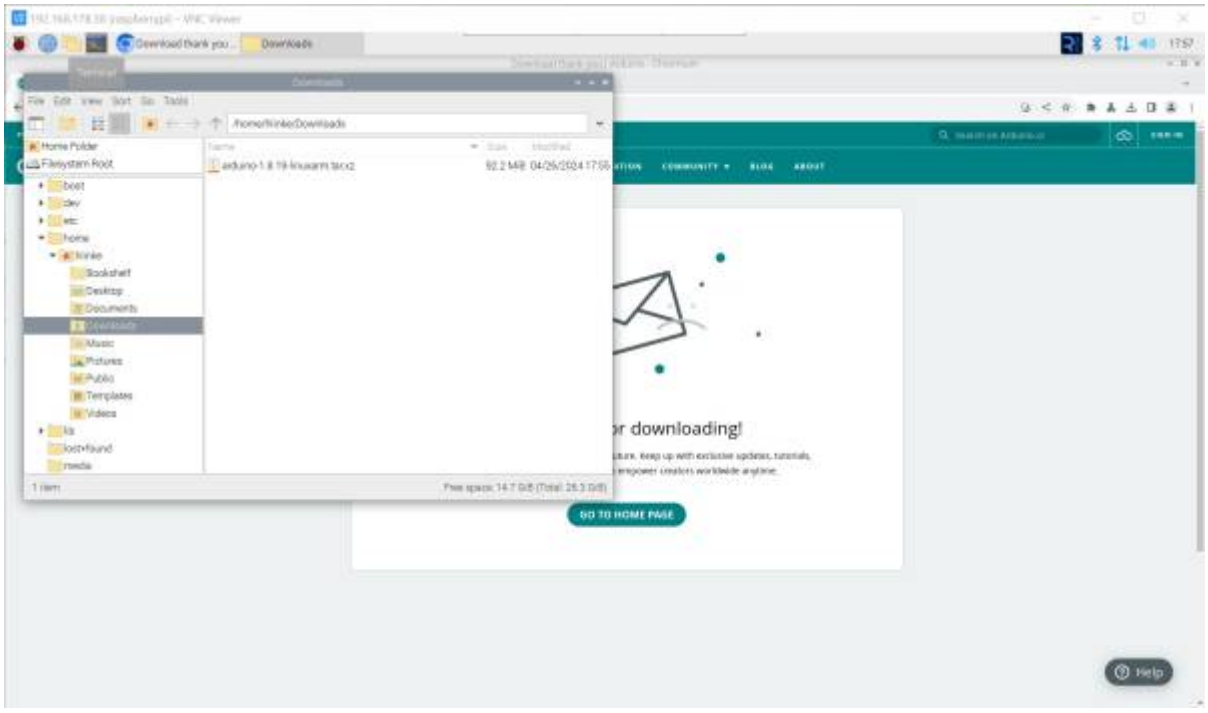
Es muß die **Version 1.8.19** verwendet werden. ACHTUNG: Die 2.xx Version geht leider nicht.



Für den Rechner die passende Version **LINUX 64 bits** auswählen und anklicken.

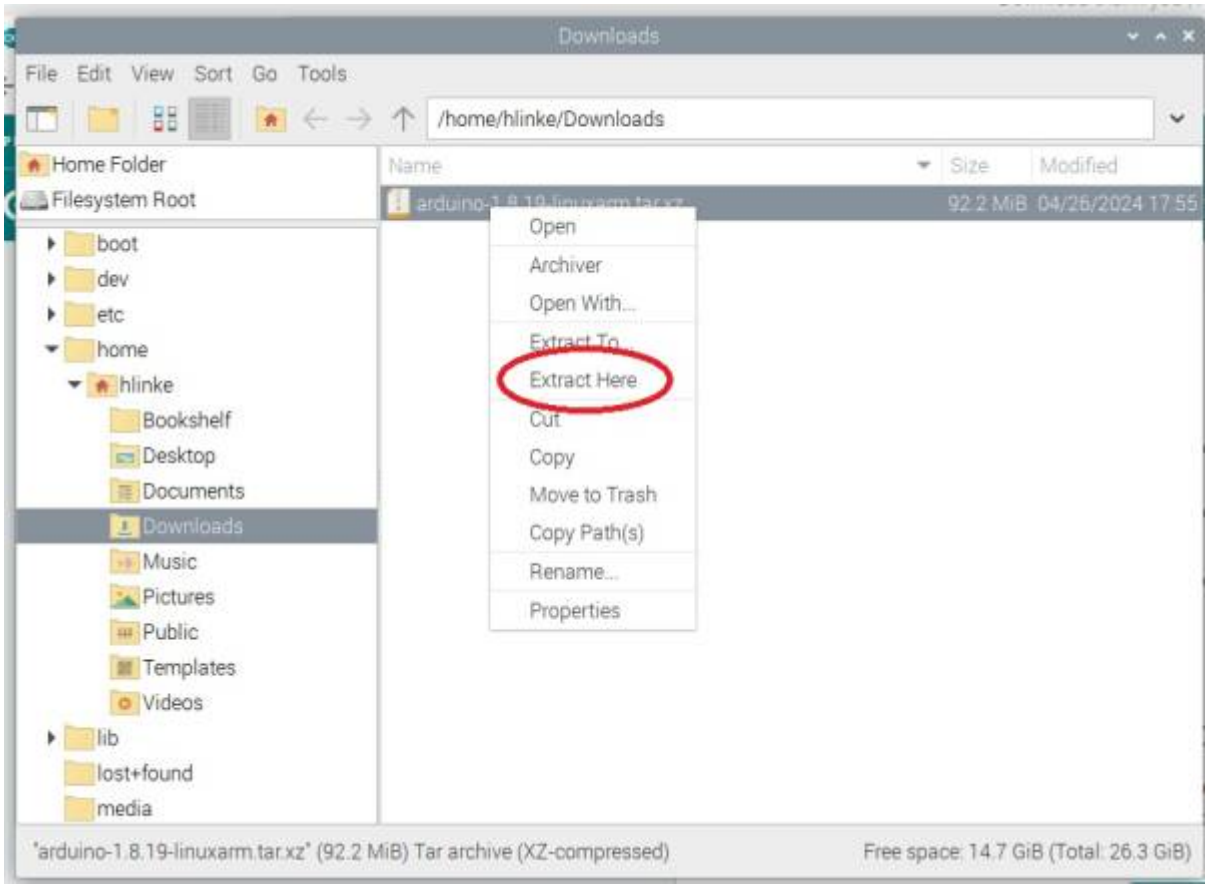


Anschließend wird man aufgefordert, für das Arduino Projekt zu spenden. Es ist gut, wenn man das großartige Projekt auf diese Weise unterstützt. Das Programm kann aber problemlos auch ohne Spenden heruntergeladen werden. Das heruntergeladene Programm, hier „arduino-1.8.19-linux64.tar.gz“ findet man in dem „Downloads„-Ordner des Rechners.

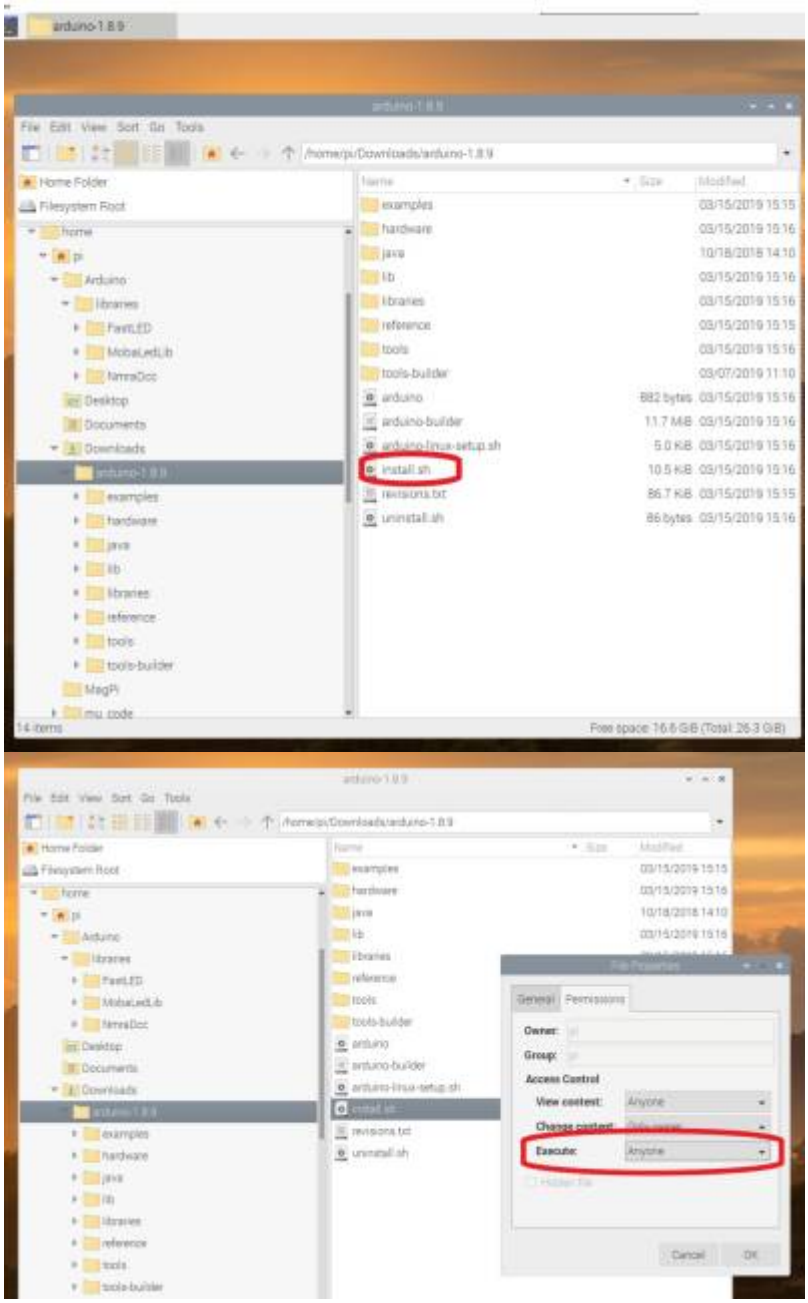


Zur Installation wird das Programm entpackt. Ich habe es gleich im Downloads Ordner entpackt. Ist nicht optimal, aber funktioniert.

Auf den Dateinamen Rechts-klicken und „Extract Here“ auswählen.



Nach dem Entpacken in das entpackte Verzeichnis wechseln und die Datei install.sh doppelt anklicken. Es kann sein, dass die Datei nicht ausgeführt werden kann. Dann auf die Datei Rechts-klicken. Die Eigenschaften (properties) auswählen. Und bei „Execute“ „Anybody“ auswählen.



Das Installationsscript erzeugt ein Desktop Icon für die ARDUINO Benutzeroberfläche.

Schritt 2: Die MobaLedLib installieren

Die Installation der MobaLedLib über die ARDUINO IDE erfolgt genauso, wie bei der original MLL, wie es hier beschrieben ist:

Anleitung für den ersten Schnelleinstieg

Die ARDUINO IDE aufrufen durch Doppelklick auf das ARDUINO-Icon auf dem Desktop.

Nun den Arduino an einen freien USB-Port anschließen. Linux installiert nun bei erstmaliger Verbindung USB/Geräte-Treiber für den Arduino. Bei fehlerfreier Installation sollten Serielle Schnitstelle und Arduino, wie im folgenden Bild dargestellt, aufrufbar sein. Die Bezeichnung der

Seriellen-Schnittstelle ist abhängt von der Belegung der Schnittstellen des Raspi mit anderen Geräten.

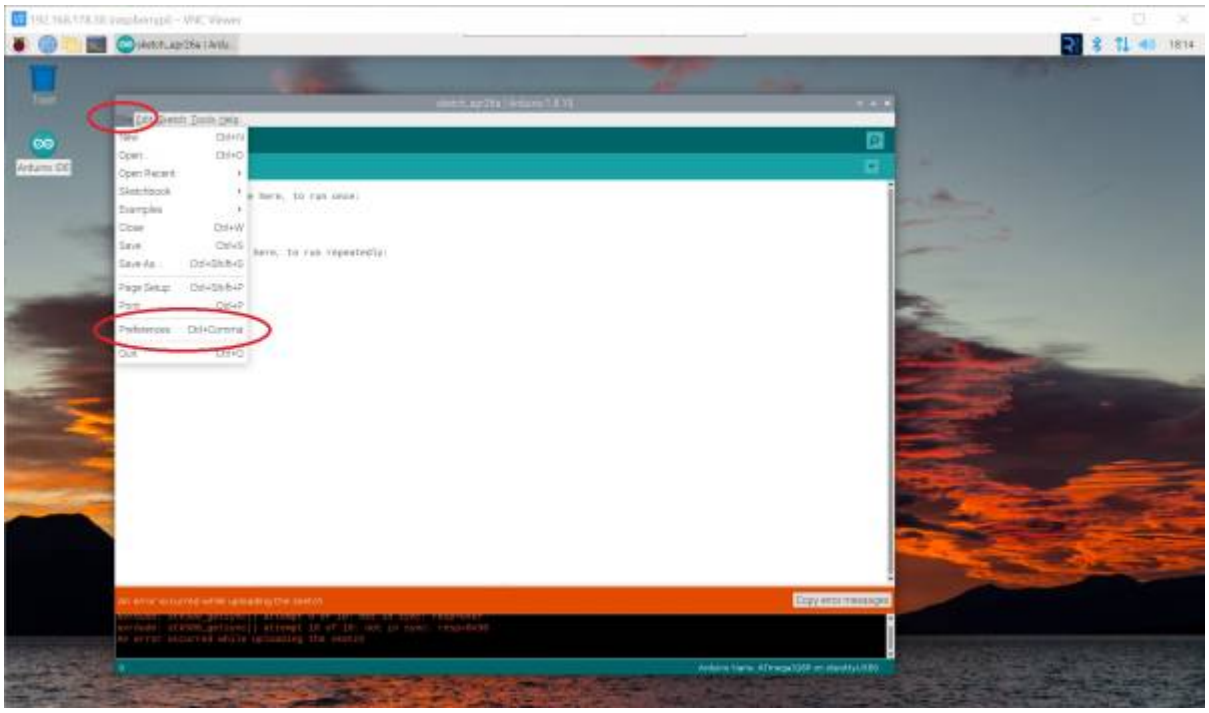
Für Einsteiger ist möglicherweise ein kleiner zusätzlicher Test interessant. Mehr dazu am Ende der Anleitung unter Test.

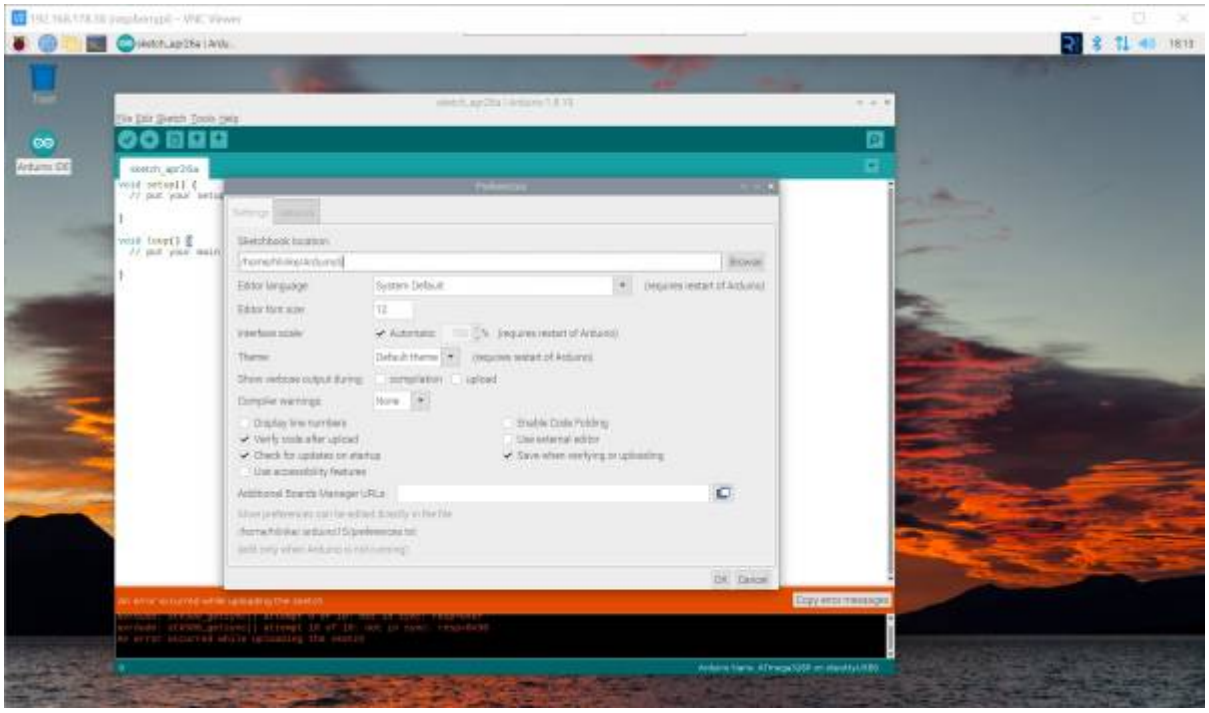
Überprüfung Sketch-Ordner

Damit alle Funktionen der MobaLedLib aufgerufen werden können, muss der Pfad zum Sketch-Ordner auf dem Standard-Pfad stehen.

```
/home/<user>/Arduino
```

Dies kann über das Menü innerhalb der Arduino IDE gemacht werden.



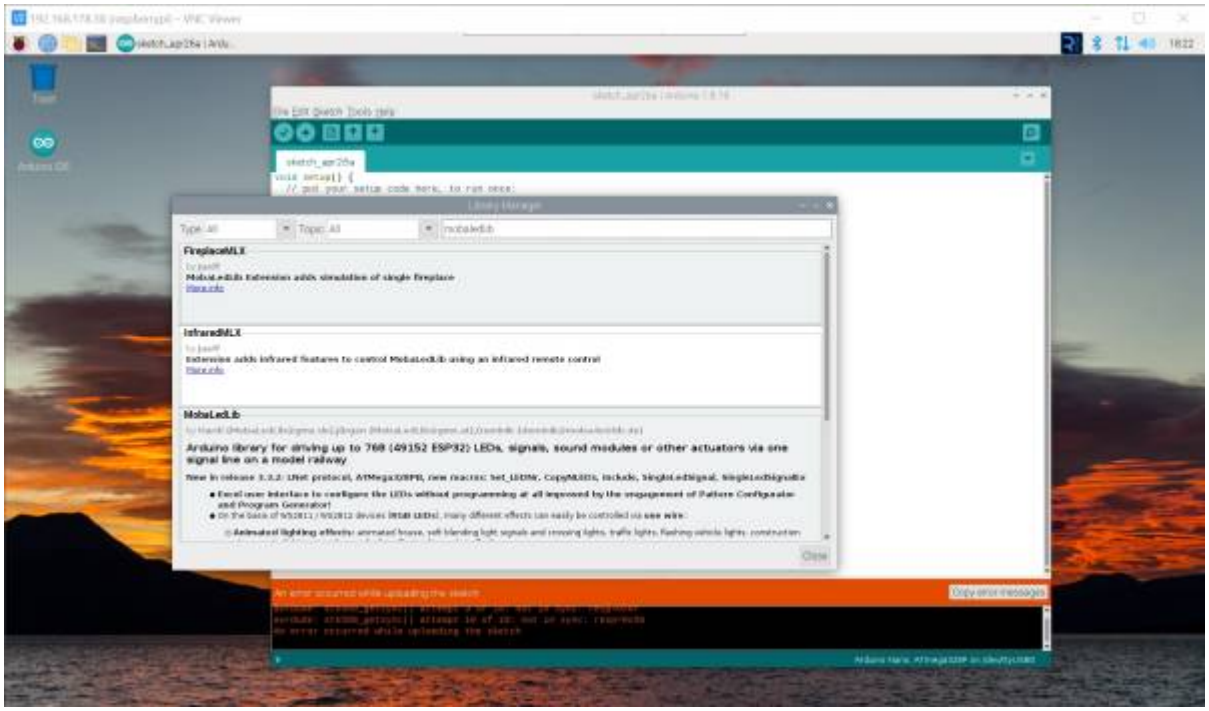


Zum Installieren der MLL in der ARDUINO IDE Tools - Manage Libraries aufrufen:

Es öffnet sich der Library Manager (der Start kann länger dauern, also Geduld ...).

Im Filter **MobaLedLib** eingeben und warten ...

Nach einiger Zeit erscheint die Beschreibung der MobaLedLib



Etwas nach unter scrollen, bis die „install“ Schaltfläche sichtbar ist und anklicken.

Es kommt die Frage ob die zusätzlichen Bibliotheken installiert werden sollen.

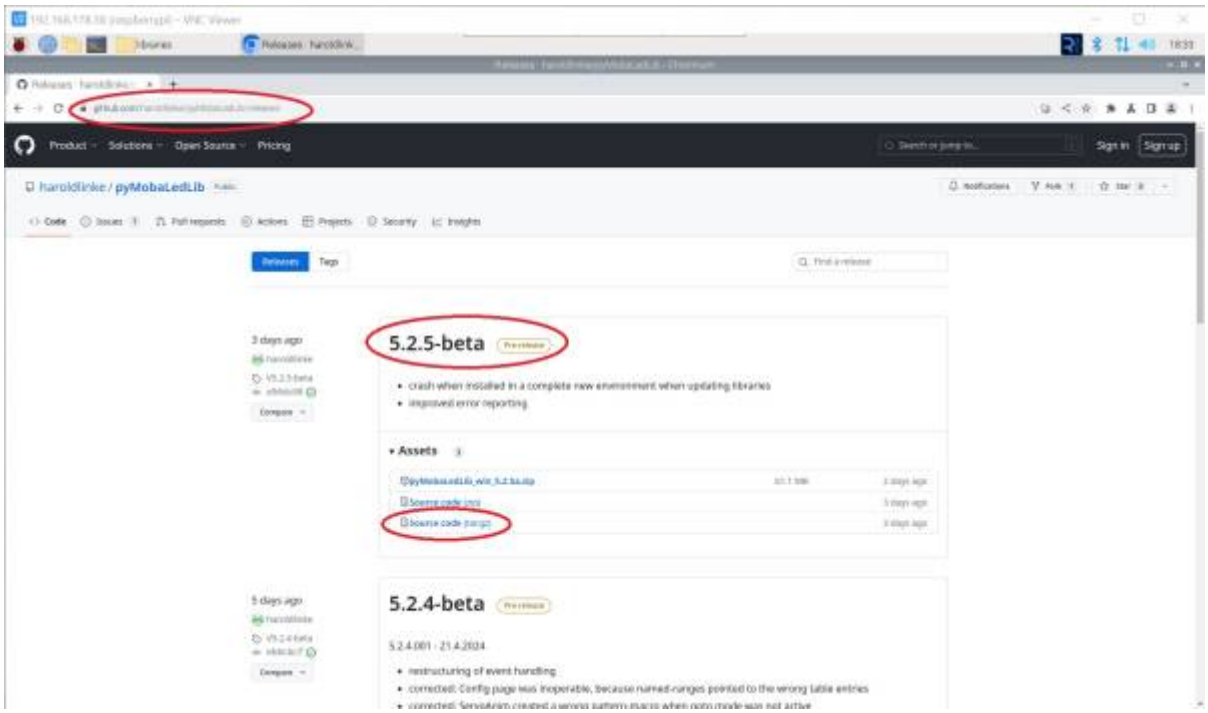


Diese Frage mit „Install all“ beantworten.

Jetzt ist die ARDUINO und MLL Installation beendet. Normalerweise wird jetzt mit dem Excel-Programm weiter gemacht. Da Excel unter Linux aber nicht läuft, benötigen wir jetzt die pyMobaLedllb.

3. PyMobaLedLib installieren

Den neuesten Release der PyMobaLedLib von GitHub https://github.com/haroldlinke/pyMobaLedLib/releases herunterladen.



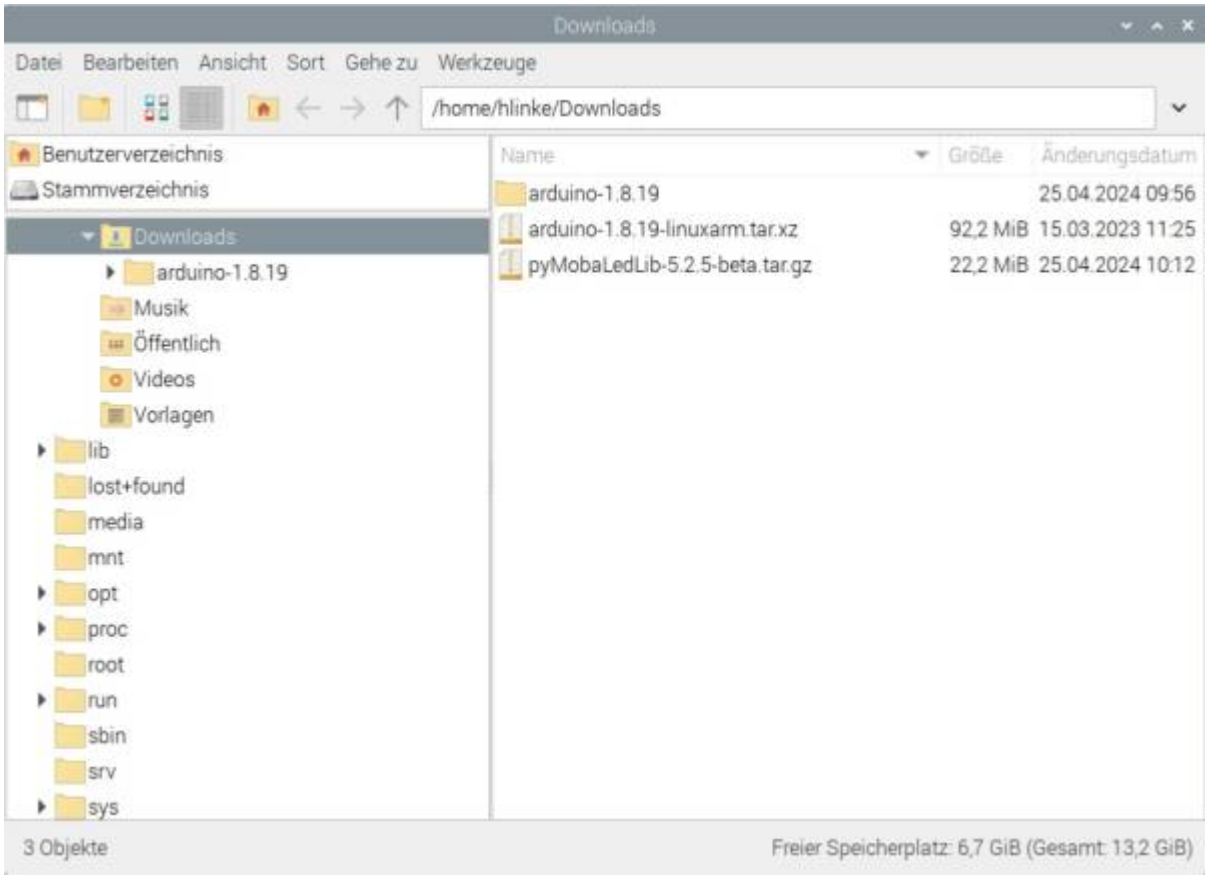
Hier ist die neueste Version die 5.2.5-beta.

Auf **Assets** klicken um die zu dem Release gehörenden Dateien zu sehen.

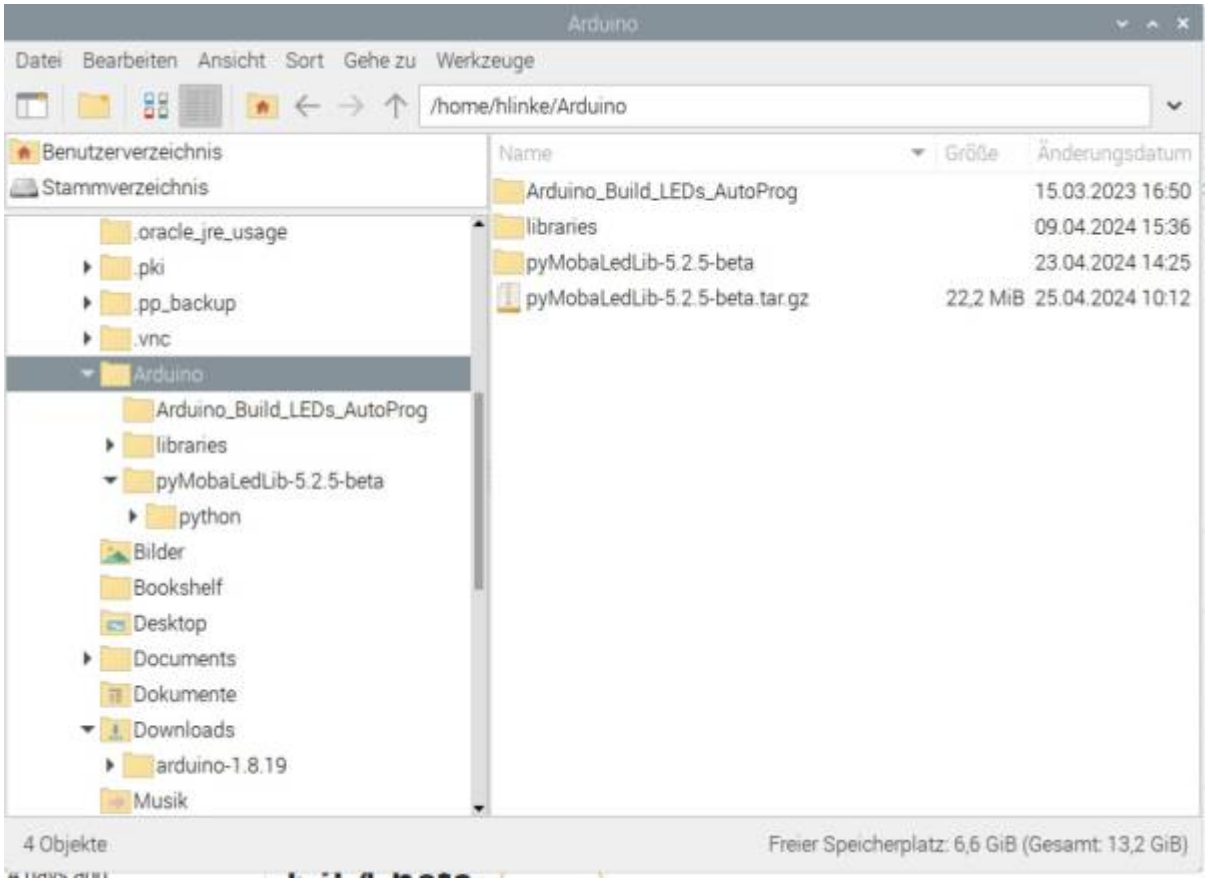
Für Linux benötigen wir die Souce code Datei vom Typ **tar.gz**

Für den Download auf den Link klicken.

Die Datei wird ins Downloadsverzeichnis heruntergeladen.



Von dort die Datei in das ARDUINO-Daten Verzeichnis kopieren. Bei einer Standard-Installation der ARDUINO SW sollte dies das Verzeichnis: home/<user>/Arduino sein. Siehe Bild:



Die ZIP-Datei „hier entpacken“, so daß sich das Verzeichnis pyMobaLedLib-xxxx in dem ARDUINO-Verzeichnis befindet. Siehe Bild oben.

Dieses Verzeichnis muss in „pyMobaLedLib“ umbenannt werden.

Als nächstes muss eine virtuelle Umgebung eingerichtet werden. Dies erfolgt über folgende Befehle Es muß ein eigenes virtuelles Python Environment für die pyMLL erstellt werden.

Dazu benötigt man das Tool venv.

Installation unter Mint:

```
sudo apt install python3-venv python3-pip
```

Achtung: Rootpasswort wird abgefragt.

Danach erstellen des virtuellen environments:

```
python3 -m venv ~/pyMLL-env
```

Es wird ein sogenanntes virtuelle Umgebung für die pyMLL geschaffen, mit eigenem Python3 programm und eigener Modulbibliothek.

Die pyMobaledLib kann dann mit dem Kommando: python3 /home/<user>/Arduino/pyMobaLedLib/python/pyMobaLedLib.py gestartet werden.

4. Desktop Icon erstellen

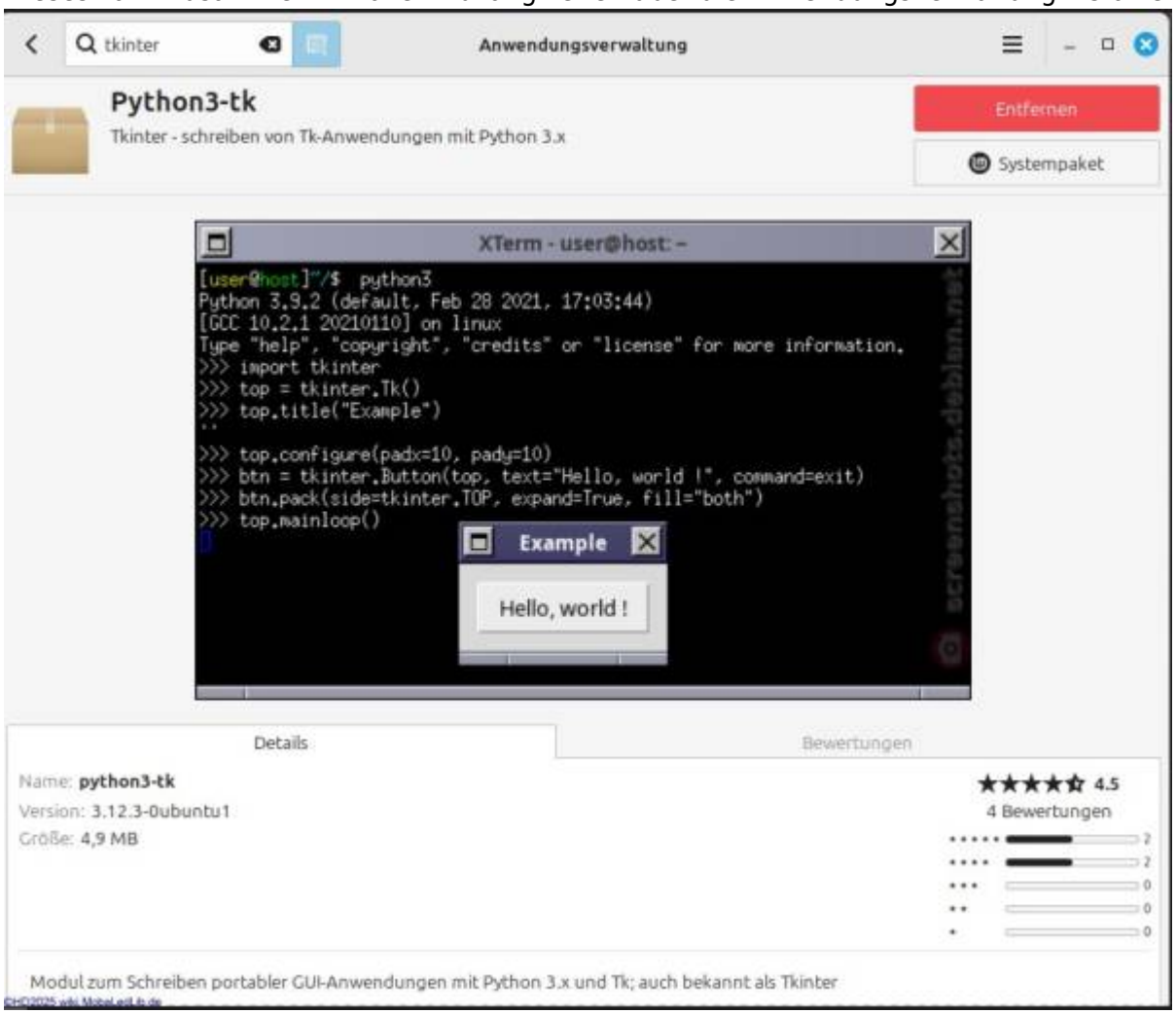
Da dieser Aufruf doch etwas kompliziert ist, kann man auf dem Desktop ein Icon installieren.

Wenn Sie ein Desktop Icon zum Aufruf der pyMobaledLib haben möchten, müssen Sie folgendermaßen vorgehen:

1. Rechtsklick auf den Desktop
2. „Neuen Starter hier erstellen“ auswählen
3. Bei Name „pyMobaLedLib“ eintragen
4. als Befehl kommt rein *python3 /home/<BENUTZERNAME>/Arduino/pyMobaLedLib/python/pyMobaLedLib.py* \\Dabei bitte <BENUTZERNAME> durch den passenden Namen ersetzen
5. Das Logo kann durch anklicken des Logos geändert werden. Das MLL-Logo ist im Pfad /home/<BENUTZERNAME>/Arduino/pyMobaLedLib/python/images/ gefunden werden

5. fehlende Pakete nachinstallieren

Für die pyMobaLedLib ist das Paket „Python3-tk“ erforderlich.
Dieses kann zusammen mit allen Abhängigkeiten über die Anwendungsverwaltung installiert werden.



Außerdem müssen folgende Pakete, ebenfalls über die Anwendungsverwaltung installiert werden:

1. „Python-is-python3“ (sollte diese Paketinstallation nicht funktionieren kann folgendes Kommando im Terminal helfen: sudo apt install python-is-python3)
2. „Python3-serial“ (In einigen Linux-Distributionen ist das Paket Python-serial nicht vorhanden, das diese Komponente schon mitgeliefert wird und installiert ist)

Dazu gibt man am Besten in dem Suchfeld „python“ ein und wählt dann aus der Liste die beiden Pakete aus.

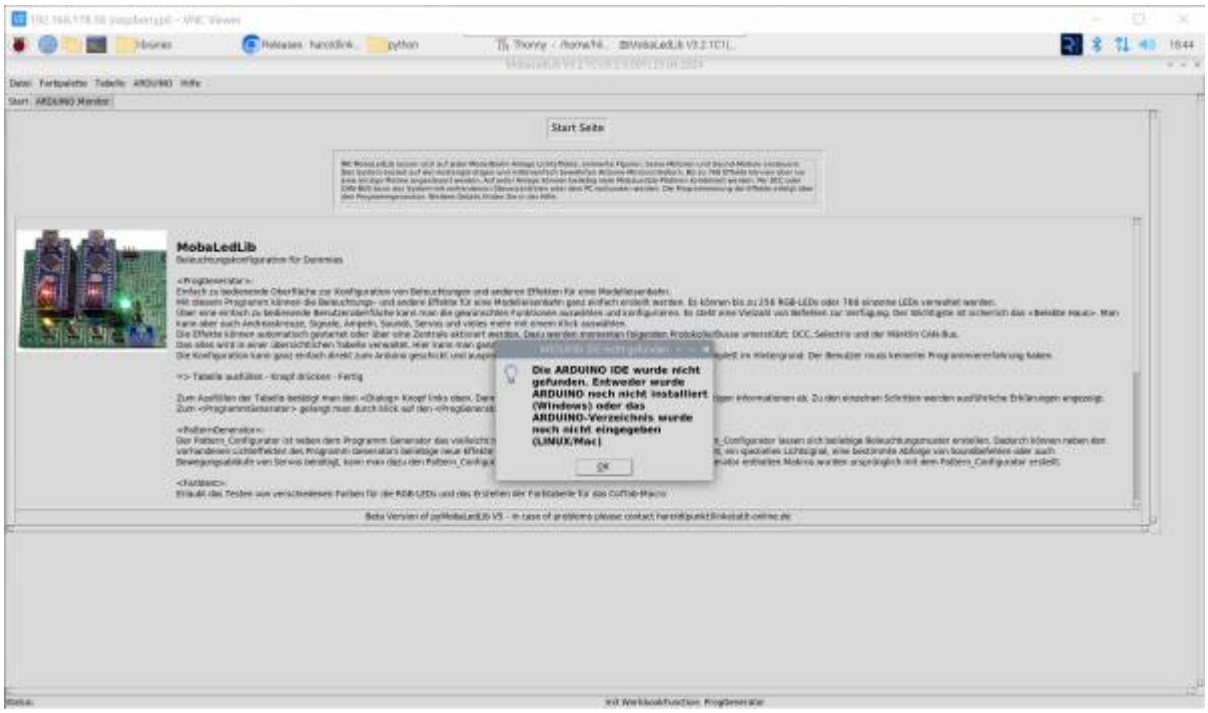
6. Berechtigungen zuweisen

Für den Zugriff auf die USB-Schnittstelle des Arduinos, benötigt der Benutzer noch die Rechte für die Nutzung.
Diese kann über das Terminal zugewiesen werden.

1. unten links auf das schwarze Symbol klicken
2. den Befehl *sudo usermod -a -G dialout <BENUTZERNAME>* eingeben. Dabei bitte wieder den Platzhalter <BENUTZERNAME> durch den korrekten Namen ersetzen.
3. Nun fragt das Fenster nach dem Kennwort für den Befehl.
4. Im Anschluss den Rechner neu starten, damit die Berechtigung korrekt geladen wird.

7. Der erste Aufruf von pyMobaLedLib

Beim ersten Aufruf meldet sich das Programm mit einer fehlermeldung: Die ARDUINO Bilbilothek wurde nicht gefunden.



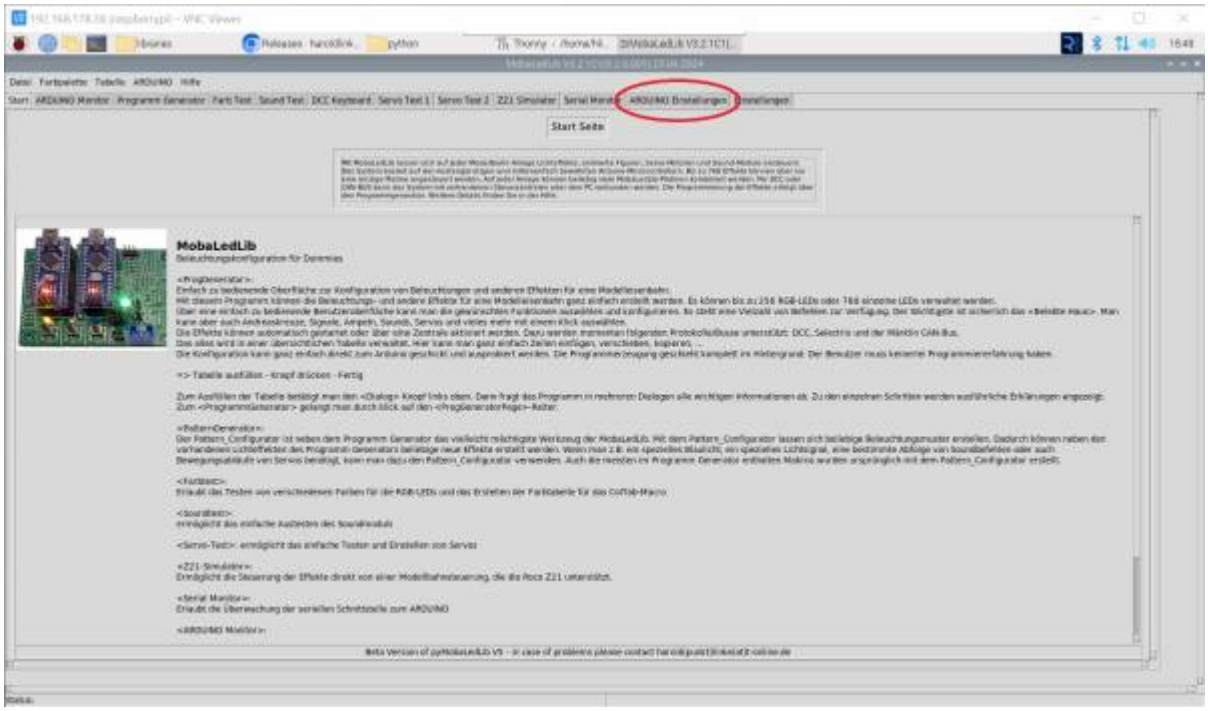
Das ist bei Linux normal, da es leider keine automatische Möglichkeit gibt, den Installationsort der ARDUINO IDE zu bestimmen. Diese Information müssen wir der pyMobaledLib von Hand mitteilen.

Zuerst diese Meldung mit „OK“ bestätigen.

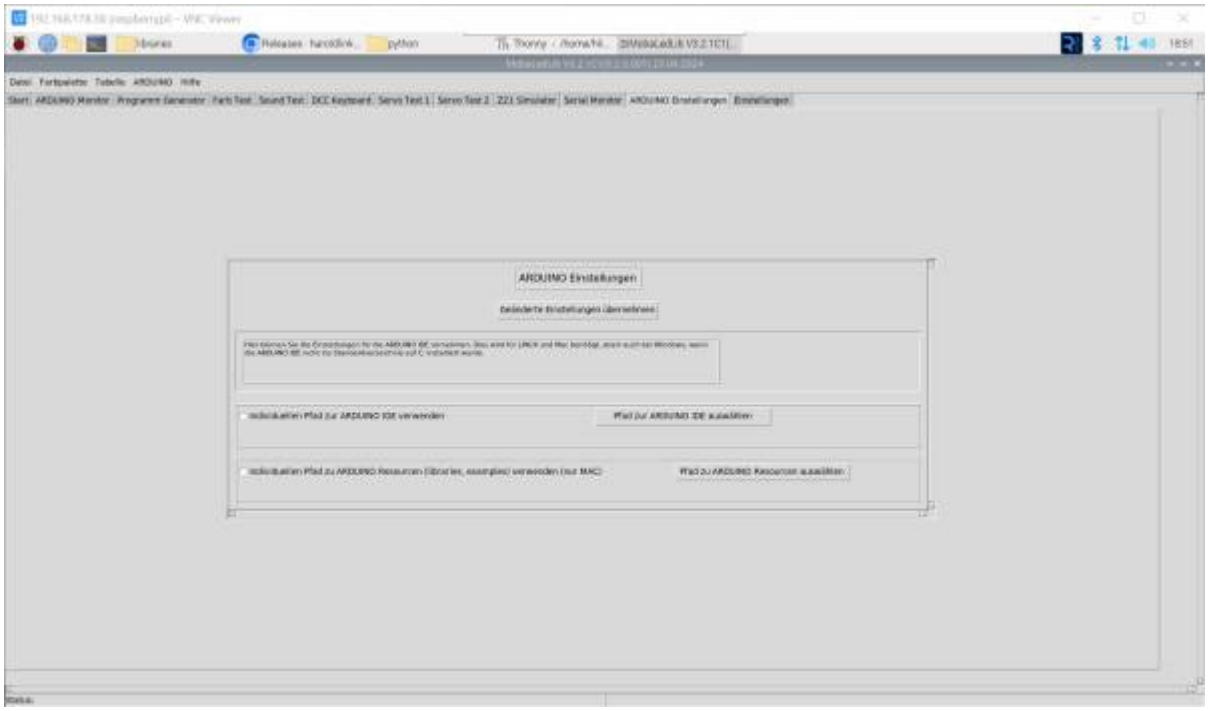
Dannach kommt eine weitere Meldung, dass das Programm eine Verbindung zum ARDUINO aufbauen möchte.

Zu dieser Meldung sagen wir „Nein“.

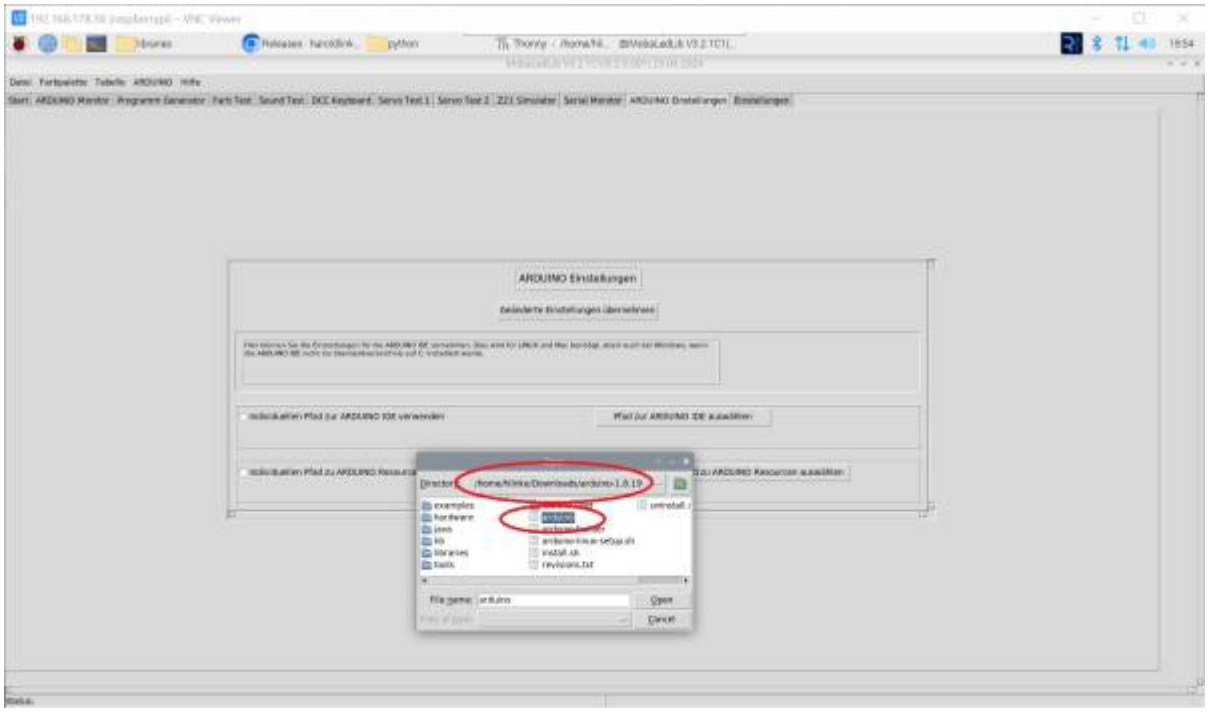
Danach sehen wir die Startseite der pyMobaLedLib.



Die Seite mit den **ARDUINO Einstellungen** öffnen.

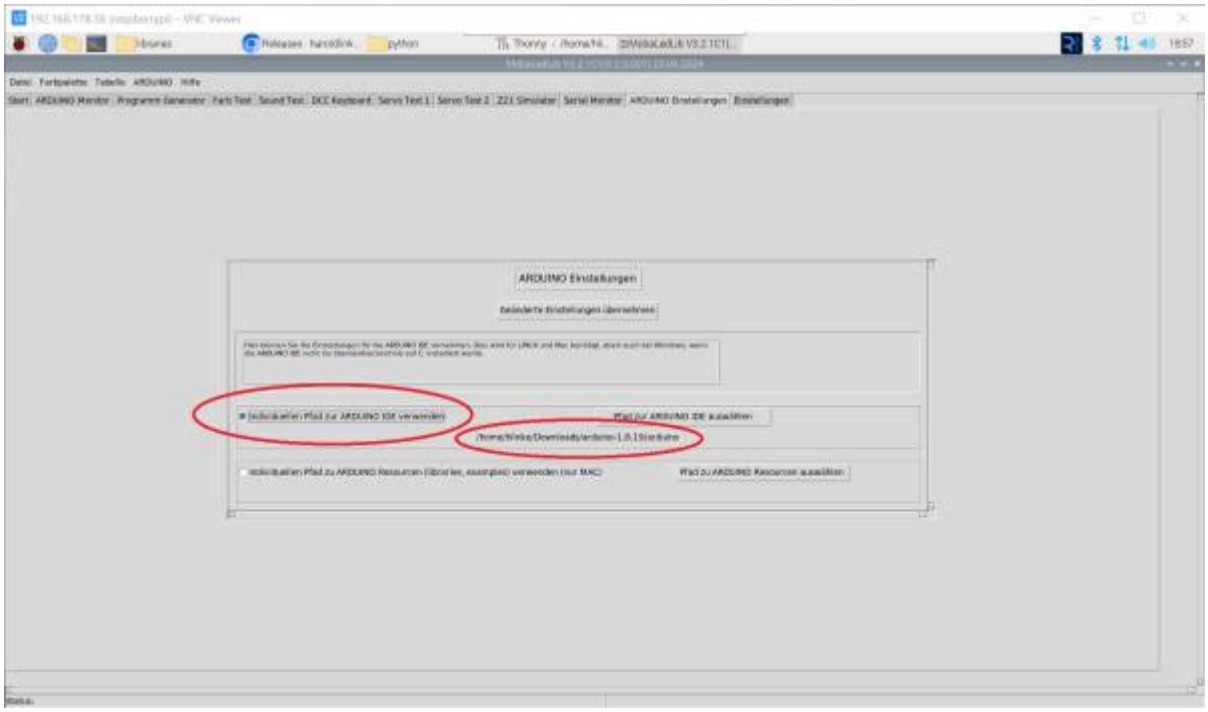


Auf den Button “**Pfad zur ARDUINO IDE auswählen**„ klicken.



Die ARDUINO IDE befindet sich in dem Ordner, in dem wir vorher die ARDUINO IDE-Datei entpackt haben. In unserem Fall /home/<user>/Downloads/arduino-1.8.19

Nicht vergessen, den Haken bei "**Individuellen Pfad zur ARDUINO IDE verwenden**„ zu setzen.

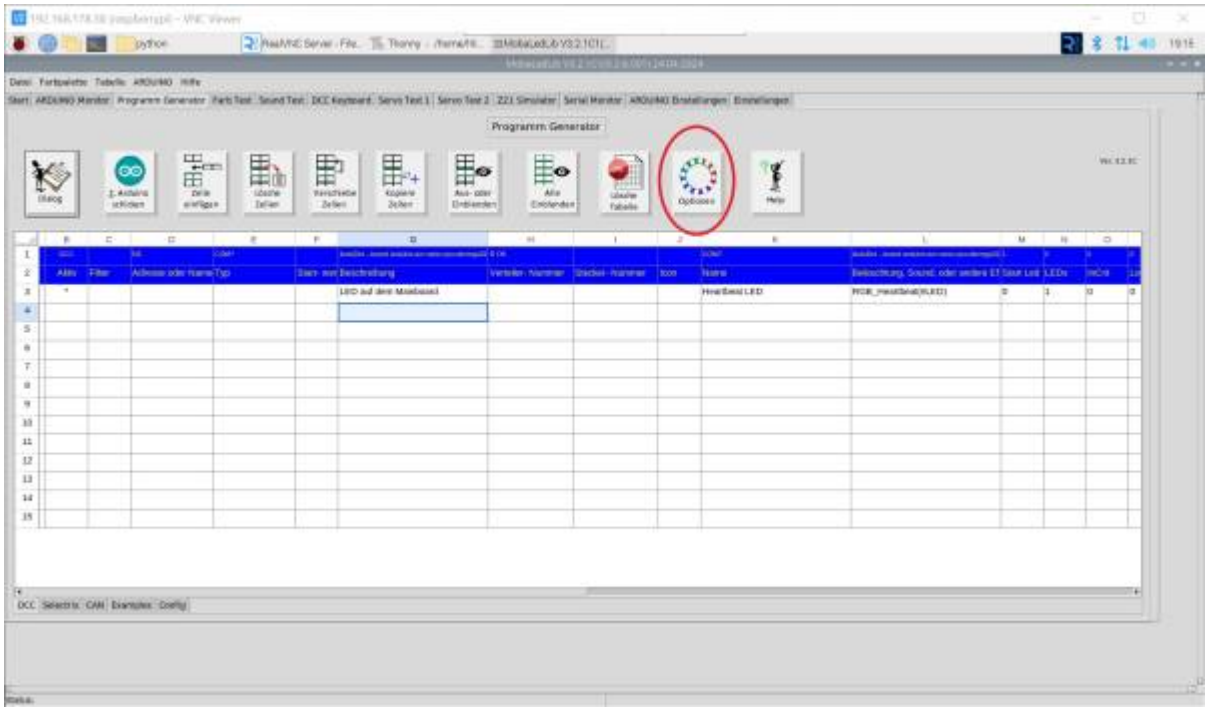


Am Besten die pyMobaLedLib jetzt neu starten.

Nach dem Start kann es sein, daß das Programm weitere Bibliotheken installieren möchte. Diese Frage mit „Ja“ beantworten.

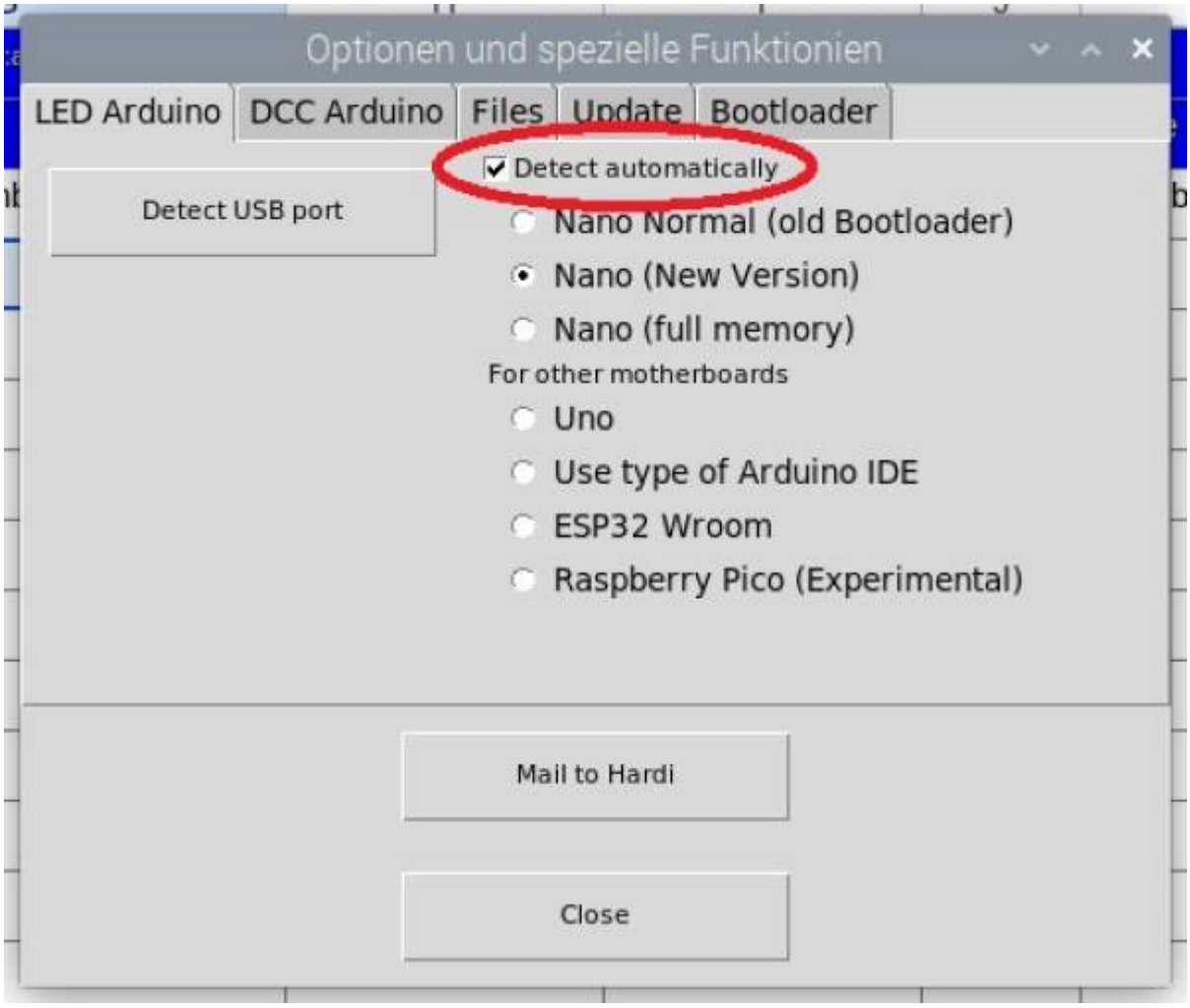
Die Frage nach der Verbindung mit dem ARDUINO sollte wieder mit „Nein“ beantwortet werden, da das Programm unter Linux den Boardtyp nicht automatisch erkennen kann und dieser sehr wahrscheinlich falsch eingestellt ist. Wir müssen den Boardtyp von Hand einstellen.

Dazu den Programm Generator öffnen:



Auf den Button „Optionen“ klicken.

Es öffnet sich das Fenster mit dem man den Boardtyp festlegen kann.



Den Haken bei „Automatisch erkennen“ löschen und den Boardtyp einstellen. Beim nano gibt es verschiedene Typen:

- neue offizielle Nanos haben meist den Typ Nano (New Version)
- China Nachbauten den Typ Nano Normal (old Bootloader)

Sie können die verschiedenen Typen auch durch probieren und sehen ob es klappt.

Nach der Einstellung des Boardtyps, den Button “**Detect USB Port**„ anklicken:

Es erscheint der USB-Port Auswahl Dialog



Hier den Port auswählen, bei dem der Nano erkannt wird entweder mit einem Namen oder als USB Serial:



Mit OK bestätigen.

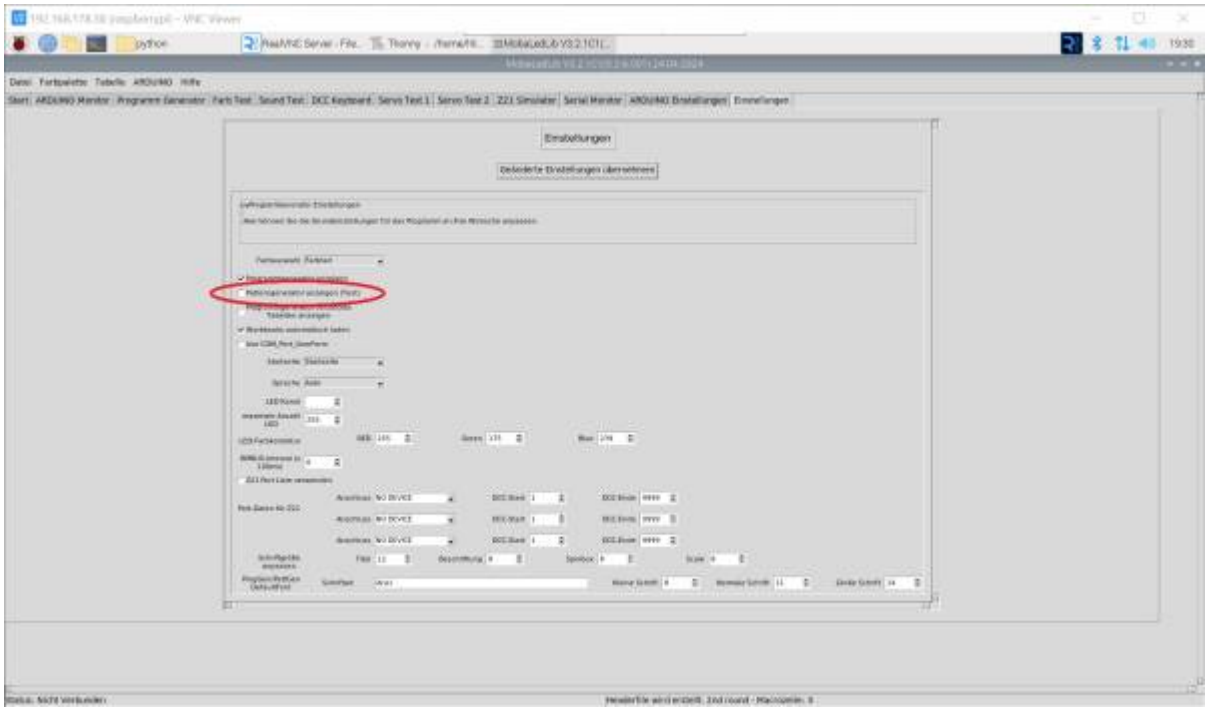
Es kommt wahrscheinlich die Meldung, dass kein ARDUINO erkannt wurde. Da die automatische Erkennung bei Linux meistens nicht funktioniert, diese Meldung mit „Nein“ ignorieren.

Dieselbe Meldung kommt auch beim Hochladen eines Programms zum ARDUINO. Diese dann auch ignorieren.

Sollte das Hochladen nicht klappen, liegt es meistens daran, dass der falsche Typ eingestellt wurde. Die obige Prozedur dann bitte wiederholen und den richtigen Typ einstellen.

Pattern Configurator aktivieren

Wenn die **Pattern Configurator** Seite nicht angezeigt wird, kann sie einfach durch eine Einstellung auf der “**Einstellungen**„ Seite aktiviert werden.



Bei der Option „Patterngenerator anzeigen muß ein Haken sein, damit die Pattern Configurator Seite angezeigt wird. Nach dem Setzen des Hakes, die Einstellungen speichern und pyMobaledLib neu starten.

From:
https://wiki.mobaledlib.de/ - **MobaLedLib Wiki**

Permanent link:
https://wiki.mobaledlib.de/anleitungen/spezial/pyprogramgenerator/linux_mint_installation?rev=1757587285

Last update: **2025/09/11 10:41**